

Law Offices
Jordan and Hamburg LLP

Chanin Building
122 East 42nd Street
New York, N. Y. 10168

C. Bruce Hamburg
Frank J. Jordan

Herbert F. Ruschmann
Jacqueline M. Steady¹
Marvin Turken, P.C.
Alfred D'Andrea, P.C.²

Of Counsel
Thomas M. Furth
Lawrence I. Wechsler

Paralegal
Michelle C. Ramos

¹Pa. Bar only
²Va. and D.C. Bars only

Assistant Commissioner for Patents
United States Patent and Trademark Office
Washington, D.C. 20231



Telephone (212) 986-2340
Facsimile (212) 953-7733

February 12, 2001

Patents, Trademarks
and Copyrights

email: jandh@attorneys.com
jandh@iplaw-worldwide.com

www.iplaw-worldwide.com

Telex 237057 JAH UR

Cable Address: PATENTMARK

Washington Office
Suite 520
2361 Jefferson Davis Highway
Arlington, Virginia 22202

Re: Application of : Kenji SHIGEKI et al.
Serial No. : 09/772,027
Filed : January 29, 2001
For : LOGIC INTEGRATED CIRCUIT, AND RECORDING
MEDIUM READABLE BY A COMPUTER, WHICH
STORES A SOURCE OF CPU CORE ON SAID LOGIC
INTEGRATED CIRCUIT
Our Ref. : F-6810

Sir:

A right of priority under 35 U.S.C §119 is hereby claimed based on applicant's following
corresponding foreign application(s):


<u>Country</u>	<u>No.</u>	<u>Filing Date</u>
Japan	2000-024826	January 28, 2000

A certified copy of said foreign application is annexed hereto.

Respectfully submitted,

JORDAN AND HAMBURG LLP

By


C. Bruce Hamburg
Reg. No. 22,389
Attorney for Applicants

CBH/ss
Enc.

Certificate of Mailing Under 37 CFR 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service
as first class mail in an envelope addressed to ASSISTANT COMMISSIONER FOR PATENTS,
WASHINGTON, DC 20231 on February 12, 2001

C. Bruce Hamburg
(Name)


(Signature)

12
2/12/01

Jordan and Hamburg
F. 6810
Kenji SHIGEKI et al.
09/772,027



日本国特許庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日
Date of Application:

2000年 1月28日

出願番号
Application Number:

特願2000-024826

願人
Applicant(s):

株式会社ローラン
株式会社ダイヘン

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年 1月26日

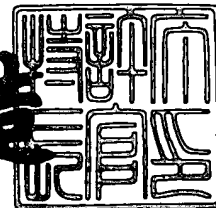
特許庁長官
Commissioner,
Patent Office

及

川

耕

造



出証番号 出証特2001-3001092

【書類名】 特許願

【整理番号】 11037H

【提出日】 平成12年 1月28日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 15/78

【発明の名称】 論理集積回路及びそのCPUコアのソースを記録したコンピュータ読み取り可能な記録媒体

【請求項の数】 7

【発明者】

 【住所又は居所】 大阪市北区東天満 1 丁目 4 番 1 6 号 株式会社ローラン
内

 【氏名】 茂木 建二

【発明者】

 【住所又は居所】 大阪市淀川区田川 2 丁目 1 番 1 1 号 株式会社ダイヘン
ン内

 【氏名】 田中 良平

【発明者】

 【住所又は居所】 大阪市北区東天満 1 丁目 4 番 1 6 号 株式会社ローラン
内

 【氏名】 中尾 俊充

【特許出願人】

 【識別番号】 595107944

 【氏名又は名称】 株式会社ローラン

 【代表者】 中尾 俊充

【特許出願人】

 【識別番号】 000000262

 【氏名又は名称】 株式会社ダイヘン

 【代表者】 西松 大三

【代理人】

【識別番号】 100084375

【弁理士】

【氏名又は名称】 板谷 康夫

【手数料の表示】

【予納台帳番号】 009531

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 不要

【書類名】 明細書

【発明の名称】 論理集積回路及びそのCPUコアのソースを記録したコンピュータ読み取り可能な記録媒体

【特許請求の範囲】

【請求項1】 CPUコアを有するフィールド・プログラマブル・ゲート・アレイ等の論理集積回路において、

前記CPUコアは、レジスタ、メモリ、及びこれらの制御部を備え、

前記CPUコアは、マイクロコードを含む命令を保有し、

前記制御部は、前記レジスタやメモリに対するイネーブル信号出力用の制御線を有し、前記命令を読み込み、この命令に含まれるマイクロコードを構成する各ビットのオン／オフ情報を前記制御線を介して各ビットに割り当てられたレジスタやメモリに伝達して、これらレジスタやメモリの状態を制御するようにしたこととを特徴とする論理集積回路。

【請求項2】 前記CPUコアは、前記メモリに対するアクセスの際のアドレスポインタ用のレジスタを有し、

前記制御部は、前記アドレスポインタ用レジスタへのインクリメント又はデクリメント指示用の制御線を有し、前記命令を読み込み、この命令に含まれるマイクロコード中の前記アドレスポインタ用レジスタのインクリメント又はデクリメント指示用のビットのオン／オフ情報を前記指示用制御線を介して前記アドレスポインタ用レジスタへ伝達し、

前記アドレスポインタ用レジスタは、前記制御部から前記指示用制御線を介してインクリメント又はデクリメント指示用のビットのオン情報を受け取った場合に、保持しているアドレスの値をカウント・アップ又はカウント・ダウンさせることを特徴とする請求項1に記載の論理集積回路。

【請求項3】 CPUコアを有するフィールド・プログラマブル・ゲート・アレイ等の論理集積回路において、

前記CPUコアは、プログラム格納用メモリ、データ格納用メモリ、及びこれらのメモリとCPUコア全体との制御を行う制御部を備え、

前記プログラム格納用メモリ及びデータ格納用メモリは、完全同期式メモリで

あり、

前記制御部は、3 段パイプライン構成での並列処理を行い、通常のクロックよりも高速のクロックを使用して、前記プログラム格納用メモリ及びデータ格納用メモリからのデータを読み込むことを特徴とする論理集積回路。

【請求項 4】 前記 CPU コアは、汎用レジスタをさらに備えると共に、メモリデータ入出力用のマルチプレクサを介することなく、前記汎用レジスタへのデータ入出力用のマルチプレクサのみを介して該汎用レジスタとメモリとを接続するデータ伝送経路を持ち、このデータ伝送経路を使用して前記メモリからのデータを前記汎用レジスタに入力することを特徴とする請求項 1 又は請求項 3 に記載の論理集積回路。

【請求項 5】 前記 CPU コアは、汎用レジスタをさらに備えると共に、ALU を介することなく、メモリデータ入出力用のマルチプレクサのみを介して該汎用レジスタとメモリとを接続するデータ伝送経路を持ち、このデータ伝送経路を使用して前記メモリへのデータを出力することを特徴とする請求項 1 又は請求項 3 に記載の論理集積回路。

【請求項 6】 前記 CPU コアは、データの入力部と出力部が独立してアクセス可能なメモリである FIFO からのデータ入力用のインタフェースが組み込まれたマルチプレクサを有し、このマルチプレクサを介して前記 FIFO からのデータを読み込むための命令を持つことを特徴とする請求項 1 又は請求項 3 に記載の論理集積回路。

【請求項 7】 CPU コアについてのソースを記録したコンピュータ読み取り可能な記録媒体であって、

前記ソースは、請求項 1 乃至請求項 6 のいずれかに記載の論理集積回路上の CPU コアについてのハードウェア記述言語レベルのソースであることを特徴とする CPU コアのソースを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、FA 機器、通信機器、家電製品等に組み込まれるフィールド・プロ

グラマブル・ゲート・アレイ（以下、FPGAと略す）、プログラマブル・ロジック・デバイス（以下、PLDと略す）等の再プログラミング可能な論理集積回路に係わり、特に、論理集積回路上にCPUコアを構築する技術に関する。

【0002】

【従来の技術】

従来のFPGA、PLD等のユーザが手元で機能を完成させる方式の論理集積回路では、複雑な解析ロジックを構築する場合に、解析ロジックを直接ハードウェア記述言語で記述して、そのロジックに該当するハードウェア回路を作成する方法が採用されていた。これに対して、ASIC（Application Specific IC）の分野では、CPUコアを有するシステムLSIの構成を採るものが多く、複雑な解析処理については、CPUコア内部のプログラムで処理する方式が採用されている。

【0003】

【発明が解決しようとする課題】

しかしながら、上記のような従来の解析ロジックに相当するハードウェア回路を作成する方式のFPGA又はPLDでは、複雑な解析ロジックを構築する場合に、回路の規模が大きくなり、また、プロトコル等の更新の頻繁なロジックの回路を構築する場合には、ロジックを更新する度にハードウェア回路を組み替える必要が生じる。また、上記のような従来のASICに用いられているCPUコアをFPGA又はPLDに直接組み込んだ場合には、ゲートの使用率が大きくなり、また、FPGA又はPLD上におけるCPUコアの占有面積が大きくなる。さらにまた、従来のASICに用いられているCPUコアは、FPGA又はPLD上の回路における内部配線遅延特性を考慮して作成されていないため、FPGA又はPLD上に組み込まれた場合に、パフォーマンスが低下して、動作スピードが低速となるという問題があった。また、従来のCPUコアには、3段パイプライン構成上で内部のプログラムやデータの格納用のメモリを完全同期式メモリとした場合に、メモリからレジスタへのロード命令と、同じレジスタからメモリへのストア命令を連続して実行すると、メモリからレジスタへのデータの読み出しが、レジスタからメモリへのデータの書き込みに間に合わず、パイプライン・

ストールが生じるので、処理が低速になるという問題があった。

【0004】

本発明は、上述した問題点を解決するためになされたものであり、FPGAやPLD等の論理集積回路上に簡易で高パフォーマンスのCPUコアを構築することができるようにして、論理集積回路上における解析ロジックの省スペース化を図ることが可能で、ロジックを更新する度にハードウェア回路を組み替える必要のないFPGA又はPLD及びそのCPUコアのソースを記録したコンピュータ読み取り可能な記録媒体を提供することを目的とする。また、3段パイプライン構成上で完全同期式メモリを採用した場合でも、パイプライン・ストールが生じないようにして、CPUコアの処理の高速化を図ることが可能な論理集積回路及びそのCPUコアのソースを記録したコンピュータ読み取り可能な記録媒体を提供することを目的とする。

【0005】

【課題を解決するための手段】

上記目的を達成するために本発明は、CPUコアを有するフィールド・プログラマブル・ゲート・アレイ等の論理集積回路において、CPUコアは、レジスタ、メモリ、及びこれらの制御部を備え、CPUコアは、マイクロコードを含む命令を保有し、制御部は、レジスタやメモリに対するイネーブル信号出力用の制御線を有し、命令を読み込み、この命令に含まれるマイクロコードを構成する各ビットのオン／オフ情報を制御線を介して各ビットに割り当てられたレジスタやメモリに伝達して、これらレジスタやメモリの状態を制御するようにしたものである。

【0006】

上記構成においては、制御部が、命令をデコードしてレジスタやメモリに対する制御信号を生成することなく、命令中のマイクロコードを構成する各ビットのオン／オフ情報を直接用いて、各ビットに対応したレジスタやメモリの状態を制御するので、制御部の行う処理を簡略化することができる。これにより、制御部を簡易な構成とすることができる。

【0007】

また、CPUコアは、メモリに対するアクセスの際のアドレスポインタ用のレジスタを有し、制御部は、アドレスポインタ用レジスタへのインクリメント又はデクリメント指示用の制御線を有し、命令を読み込み、この命令に含まれるマイクロコード中のアドレスポインタ用レジスタのインクリメント又はデクリメント指示用のビットのオン／オフ情報を、指示用制御線を介してアドレスポインタ用レジスタへ伝達し、アドレスポインタ用レジスタは、制御部から指示用制御線を介してインクリメント又はデクリメント指示用のビットのオン情報を受け取った場合に、保持しているアドレスの値をカウント・アップ又はカウント・ダウンさせることが望ましい。このアドレスポインタ用レジスタを繰り返し利用しながらメモリに対するアクセスを行うことにより、メモリ上の連続したアドレス空間のデータ処理を効率的に行うことができる。

【0008】

また、請求項3の発明は、CPUコアを有するフィールド・プログラマブル・ゲート・アレイ等の論理集積回路において、CPUコアは、プログラム格納用メモリ、データ格納用メモリ、及びこれらのメモリとCPUコア全体との制御を行う制御部を備え、プログラム格納用メモリ及びデータ格納用メモリは、完全同期式メモリであり、制御部は、3段パイプライン構成での並列処理を行い、通常のクロックよりも高速のクロックを使用して、プログラム格納用メモリ及びデータ格納用メモリからのデータを読み込むものである。

【0009】

この構成においては、制御部は、通常のクロックよりも高速のクロックを使用して、完全同期式メモリであるプログラム格納用メモリ及びデータ格納用メモリからのデータを読み込むので、通常のクロックを使用してこれらのデータを読み込んだ場合と比べて、これらのメモリに格納された命令やデータの読み込みのタイミングを早めることができる。これにより、例えば、データ格納用メモリからレジスタへのデータの読み出し命令と、同じレジスタからデータ格納用メモリへのデータの書き込み命令が連続して実行された場合でも、データ格納用メモリからレジスタへのデータの読み出しが、後続するレジスタからデータ格納用メモリへのデータの書き込みに間に合い、従って、3段パイプライン構成上で完全同期

式メモリを採用しているにも拘わらず、パイプライン・ストールが生じないようにすることができる。

【 0 0 1 0 】

また、CPUコアは、汎用レジスタをさらに備えると共に、メモリデータ入出力用のマルチプレクサを介することなく、汎用レジスタへのデータ入出力用のマルチプレクサのみを介して汎用レジスタとメモリとを接続するデータ伝送経路を持ち、このデータ伝送経路を使用してメモリからのデータを汎用レジスタに入力することが望ましい。これにより、汎用レジスタが、メモリデータ入出力用のマルチプレクサを介することなく、汎用レジスタ用のマルチプレクサのみを介してメモリからのデータを入力することができるので、メモリから汎用レジスタへのデータの読み込み処理の高速化を図ることができる。

【 0 0 1 1 】

また、CPUコアは、汎用レジスタをさらに備えると共に、ALUを介することなく、メモリデータ入出力用のマルチプレクサのみを介して汎用レジスタとメモリとを接続するデータ伝送経路を持ち、このデータ伝送経路を使用してメモリへのデータを出力することが望ましい。これにより、汎用レジスタが、ALUを介することなく、メモリデータ入出力用のマルチプレクサのみを介してメモリへのデータを出力することができるので、汎用レジスタからメモリへのデータの書き込み処理の高速化を図ることができる。

【 0 0 1 2 】

また、CPUコアは、データの入力部と出力部が独立してアクセス可能なメモリであるFIFOからのデータ入力用のインタフェースが組み込まれたマルチプレクサを有し、このマルチプレクサを介してFIFOからのデータを読み込むための命令を持つことが望ましい。これにより、このマルチプレクサにFIFOを接続して、FIFOからのデータの読み込み命令を発することで、FIFOのデータを容易にCPUコア内に読み込むことができる。

【 0 0 1 3 】

また、請求項7の発明は、CPUコアについてのソースを記録したコンピュータ読み取り可能な記録媒体であって、そのソースを、請求項1乃至請求項6のい

ずれかに記載のフィールド・プログラマブル・ゲート・アレイ等の論理集積回路上のCPUコアについてのハードウェア記述言語レベルのソースとしたものである。この構成においては、コンピュータにソースを読み取らせることにより、上記と同様な作用を得ることができる。

【0014】

【発明の実施の形態】

以下、本発明の一実施形態による論理集積回路であるフィールド・プログラマブル・ゲート・アレイ（以下、FPGAという）について図面を参照して説明する。図1に本実施形態によるFPGAのCPUコア周辺の構成を示す。FPGA 1は、簡易なRISCプロセッサであるCPUコア2、再利用可能なマクロセルやメガセル等のIP（Intellectual Property）3、マスタ・クロック4、及びCPUコア2の実機テストを行う際にデータや制御信号の流れをコントロールするASH（Adaptive Scan Handler:適用型走査処理部）36より構成される。CPUコア2は、FPGA 1上の解析ロジックの規模を小さくするために組み込まれたIPの一種である。このCPUコア2は、命令の内容をデコードしてCPUコア2全体の制御を行うマイクロ・コントローラ5、種々のレジスタやマルチプレクサから成るデータバス6、CPUコア2の処理を記述したプログラムを格納するプログラム・メモリ7、CPUコア2が使用するデータを記憶するデータ・メモリ8、マイクロコード（レジスタ等の制御用のビット情報からなるコード）に基づいてレジスタ等を制御するマイクロコード・コントローラ9を含む。また、CPUコア2は、ASH 36からの制御信号に基づいてCPUコア2内部のレジスタ等に対するデータの入出力等を行うためのスキャン・パス10を備えている。さらに、CPUコア2は、FPGA 1の内部又は外部で発生する割り込み信号を検出してマイクロ・コントローラ5へ通知する割り込みコントローラ11、RS-232CインタフェースによりFPGA 1内部の通信を行うためのRS-232Cモジュール12、CPUコア2の内部タイマであるタイマ13、マスタ・クロック4に基づいてデータバス6内のレジスタ等に対するクロックを生成するクロック・ジェネレータ14を備えている。

【0015】

上記マイクロコード・コントローラ 9 は、プログラム・メモリ 7 に格納された命令中のマイクロコードと第 1 オペコード（図 9 参照）を読み取り、マイクロ・コントローラ 5 は、上記プログラム・メモリ 7 に格納された命令中のマイクロコード以外の部分を読み取る。そして、これらマイクロ・コントローラ 5 とマイクロコード・コントローラ 9 の各々は、それぞれの読み取った情報に基づいてデータバス 6 内の各種レジスタやマルチプレクサの制御を行う。つまり、マイクロ・コントローラ 5 とマイクロコード・コントローラ 9 よりデータバス全体の制御部 2 0 が構成される。

【 0 0 1 6 】

次に、上記データバス 6 の構成について説明する。上記データバス 6 は、算術論理演算器（Arithmetic Logic Unit:以下、ALUと略す）、プログラム・カウンタ（Program Counter:以下、PCと略す）22、フェッチ（Fetch）用の命令レジスタであるIRF（Instruction Register for Fetch）23、イグゼキュート（Execute）用の命令レジスタであるIRE（Instruction Register for Execute）24、データ・メモリ 8 等のアドレスを示すために用いられるアドレスポインタ用レジスタであるX-REG（X Register）25、Y-REG（Y Register）26、スタックポインタ（Stack Pointer:以下、SPと略す）27、汎用レジスタであるA-REG（A Register）28、B-REG（B Register）29、プログラム・メモリ 7 のアドレス用のマルチプレクサであるP-MUX（Program Counter Multiplexer）30、データ・メモリ 8 のアドレス用のマルチプレクサであるSA-MUX（Sram Address Multiplexer）31、データ・メモリ 8 に対するデータ入出力用のマルチプレクサであるDATA-MUX（Data Multiplexer）32、汎用レジスタに対するデータ入出力用のマルチプレクサであるREG-MUX（Register Multiplexer）33、ユーザがCPUコア 2 の外部に作成したFIFO（First-In First-out）18に対するデータ入出力用のマルチプレクサであるF-MUX（Fifo Multiplexer）34、IP 3 に対するデータ入出力用のマルチプレクサであるIP-MUX（IP Multiplexer）35より構成される。

【 0 0 1 7 】

上記のCPUコア 2 には、FPGA 1 に特化した配線経路の工夫がなされてい

る。何故なら、一般に A S I C やゲートアレイ (Gate Array) と比較して、F P G A や P L D (Programmable Logic Device) の場合は、内部配線遅延特性がシステムのパフォーマンスに大きな影響を与えるため、F P G A 1 上に C P U コア 2 を構築する場合には、この点を考慮する必要があるからである。以下に、C P U コア 2 上の汎用レジスタやデータ・メモリ 8 周辺の配線経路の工夫について説明する。

【 0 0 1 8 】

先ず、図 2 を参照して、データ・メモリ 8 から汎用レジスタである A - R E G 2 8 にデータをロードする際の配線経路に関する工夫について説明する。データ・メモリ 8 から A - R E G 2 8 にデータをロードする際の配線経路 (配線 P 1, P 2 より構成される配線経路) は、データ・メモリ 8 を汎用レジスタ用のマルチプレクサである R E G - M U X 3 3 のみを介して A - R E G 2 8 と結ぶものである。従来の A S I C 等の C P U コアの場合、通常、データ・メモリから汎用レジスタ (A - R E G 2 8 に相当) にデータをロードする際の配線経路は、データ・メモリのデータ入出力用のマルチプレクサ (D A T A - M U X 3 2 に相当) と汎用レジスタ用のマルチプレクサ (R E G - M U X 3 3 に相当) を介してデータ・メモリを汎用レジスタと結ぶものであった。しかし、F P G A 1 上で C P U コア 2 を構築する場合、データ・メモリ 8 からの出力データが、データ・メモリ 8 用のマルチプレクサである D A T A - M U X 3 2 と汎用レジスタ用のマルチプレクサである R E G - M U X 3 3 の 2 つのマルチプレクサを通過すると、配線遅延の時間が長くなる。従って、高速な処理を行う場合に、データ・メモリ 8 からのデータを A - R E G 2 8 に読み込むタイミング (マイクロコード・コントローラ 9 から A - R E G 2 8 の L D 端子へ出力されるイネーブル信号と、クロック・ジェネレータ 1 4 から A - R E G 2 8 へ送られるクロック信号により決定されるタイミング) に、データ・メモリ 8 から A - R E G 2 8 の D I N 端子へのデータの出力が間に合わない。このため、上記のように、データ・メモリ 8 を、データ・メモリ用のマルチプレクサである D A T A - M U X 3 2 を介することなく、汎用レジスタ用のマルチプレクサである R E G - M U X 3 3 のみを介して A - R E G 2 8 と結ぶ配線経路を設けて、この配線経路を通してデータ・メモリ 8 からのデー

データをA-REG 28に読み込むことにより、配線遅延の時間を短縮した。これにより、高速な処理を行う場合でも、データ・メモリ8からのデータをA-REG 28に読み込むタイミングに、データ・メモリ8からA-REG 28のDIN端子へのデータの出力が間に合うので、データ・メモリ8からA-REG 28へのデータの読み込み処理（ロード処理）の高速化を図ることができる。上記の説明では、汎用レジスタとしてA-REG 28を用いた場合について説明したが、汎用レジスタとしてB-REG 29を用いた場合も同様である。

【0019】

次に、図3を参照して、汎用レジスタであるA-REG 28からデータ・メモリ8にデータをストアする際の配線経路に関する工夫について説明する。A-REG 28からデータ・メモリ8にデータをストアする際の配線経路（配線P3，P4より構成される配線経路）は、A-REG 28をデータ・メモリ用のマルチプレクサであるDATA-MUX 32のみを介してデータ・メモリ8と結ぶものである。従来のASIC等のCPUコアの場合、通常、汎用レジスタ（A-REG 28に相当）からデータ・メモリにデータをストアする際の配線経路は、汎用レジスタをALUとデータ・メモリ用のマルチプレクサ（DATA-MUX 32に相当）とを介してデータ・メモリと結ぶものであった。しかし、FPGA1上でCPUコア2を構築する場合、汎用レジスタ（A-REG 28）からの出力データが、ALU 21を通過すると、配線遅延の時間が長くなる。従って、高速な処理を行う場合に、データ・メモリ8の書き込みタイミング（マイクロコード・コントローラ9からデータ・メモリ8のWR端子へ出力されるWE（Write Enable）信号と、クロック・ジェネレータ14からデータ・メモリ8へ送られるクロック信号とにより決定されるタイミング）に、A-REG 28からデータ・メモリ8のDIN端子へのデータの出力が間に合わない。このため、上記のように、A-REG 28をALU 21を介することなく、DATA-MUX 32のみを介してデータ・メモリ8と結ぶ配線経路を設けて、この配線経路を通してA-REG 28からのデータをデータ・メモリ8に書き込むことにより、配線遅延の時間を短縮した。これにより、高速な処理を行う場合でも、A-REG 28からデータ・メモリ8のDIN端子へのデータの出力が、データ・メモリ8の書き込みタ

イミングに間に合うので、A-REG 28からデータ・メモリ 8へのデータの書き込み処理（ストア処理）の高速化を図ることができる。上記の説明では、汎用レジスタとしてA-REG 28を用いた場合について説明したが、汎用レジスタとしてB-REG 29を用いた場合も同様である。

【0020】

CPUコア2は、FIFOからのデータ入力命令やポストインクリメント命令等のデータ転送に便利な命令を実現できる機構を採用している。先ず、図4を参照して、FIFOからのデータ入力命令を実現するための機構について説明する。上述したように、CPUコア2は、外部のFIFO 18に対するデータ入力用のマルチプレクサであるF-MUX 34を有している。このF-MUX 34には、FIFOインタフェースが組み込まれている。また、CPUコア2は、FIFO 18に対して読み込み用の制御信号を発することができる。CPUコア2は、FIFO命令の実行時に、FIFO 18に対して読み込み用の制御信号を発し、F-MUX 34に組み込まれたFIFOインタフェースを用いて、FIFO 18からのデータを汎用レジスタであるA-REG 28、B-REG 29に読み込む。F-MUX 34の入力ポート0～3は、FIFO 18からのデータの読み込み以外にも、外部信号の入力や外部レジスタ41等からの固定値の入力にも用いることができる。従って、CPUコア2は、FIFO命令を実行することで、外部信号のモニタ等も行うことができる。

【0021】

次に、図5を参照して、ポストインクリメント命令を実現するための機構について説明する。上述したように、CPUコア2は、データ・メモリ 8等のアドレスを示すために用いられるアドレスポインタ用レジスタであるX-REG 25を有している。このX-REG 25には、マイクロコード・コントローラ9からのインクリメント指示の受信用の制御線L5が接続されている。また、X-REG 25は、この制御線L5を介してインクリメント指示の信号が入力された場合に、内部で保持しているアドレスの値をカウント・アップさせる回路を有している。マイクロコード・コントローラ9は、プログラム・メモリ7中のポストインクリメント命令を読み込むと、この命令中のマイクロコードのインクリメント指示

用のビットのオン情報（インクリメント指示信号）を、制御線 L5 を介して X-REG25 へ伝達する。X-REG25 は、このインクリメント指示用のビットのオン情報を受け取ると、内部で保持しているアドレスの値に 1 を加算する。この X-REG25 内部のアドレスの値は、データ・メモリ 8 のアドレス入力用のマルチプレクサである S A-MUX31 を介してデータ・メモリ 8 のアドレス入力用の端子である ADR 端子に出力される。従って、ポストインクリメント命令を繰り返して用いることで、データ・メモリ 8 のアドレスに対応した X-REG25 の値を順次カウント・アップさせていくことができる。これにより、データ・メモリ 8 上の連続したアドレス空間のデータを効率的に処理することができる。

【0022】

次に、図 6 乃至図 8 を参照して、CPU コア 2 に採用されている 3 段パイプライン構成上で完全同期式メモリにアクセスする場合の処理の高速化を図るための工夫について説明する。前述した図 1 に示すように、CPU コア 2 は、プログラム格納用のプログラム・メモリ 7 とデータ記憶用のデータ・メモリ 8 とを別個に持つ。これらのメモリ 7, 8 は、共に完全同期式メモリである。何故なら、非同期式メモリの場合は、回路の大規模化・高速化に伴って、信号遅延の問題が大きくなる。すなわち、メモリの回路構成によって、メモリにアドレスを指定してからメモリからのデータが出力されるまでの時間に差がある。従って、プログラム・メモリ 7 とデータ・メモリ 8 は、両方とも完全同期式メモリであることが望ましい。しかし、3 段パイプライン構成上でこれらのメモリ 7, 8 を完全同期式メモリとすると、シングルクロックでこれらのメモリにアクセスした場合に、以下の問題が生じる。例えば、図 6 に示されるように、データ・メモリ 8 上の ADR1 のアドレスのデータを図 1 の A-REG28 にロードする命令（LDA 命令）と、このロード命令で読み出した A-REG28 のデータをデータ・メモリ 8 上の ADR2 のアドレスにストアする命令（STA 命令）とを連続して実行した場合には、プログラム・メモリ 7 からのデータの読み込みと、データ・メモリ 8 からのデータの読み込みとは、クロックの立ち下がり時にしか行えないため、ロード命令による A-REG28 の更新が、ストア命令により A-REG28 のデ

タをデータ・メモリ 8 に書き込むタイミングに間に合わない。すなわち、図 1 の制御部 20 は、STA 命令のコマンド解釈のタイミングで、データ・メモリ 8 へのデータ書き込み許可信号である WE (Write Enable) 信号を出力し、この WE 信号の出力中にクロックが立ち上がるタイミング (図中の矢印 t 8 で示されるタイミング) で、A-REG 28 のデータをデータ・メモリ 8 に書き込むが、この時点では、A-REG 28 の内容はデータ・メモリ 8 上の ADR 1 のアドレスのデータに更新されていない。従って、ストア命令の実行により、データ・メモリ 8 の ADR 2 のアドレスには、ロード命令実行前の A-REG 28 のデータが書き込まれてしまう。

【0023】

上記の問題をシングルクロックのまま解決するためには、図 7 に示されるように、上記のロード命令 (LDA 命令) とストア命令 (STA 命令) との間に NOP (No Operation) を挿入すれば、ロード命令による図 1 の A-REG 28 の更新が、ストア命令により A-REG 28 のデータをデータ・メモリ 8 に書き込むタイミング (図中の矢印 t 10 で示されるタイミング) に間に合う。しかし、このように NOP を挿入すると、CPU コア 2 の処理が低速になる。

【0024】

そこで、本実施形態による CPU コア 2 の制御部 20 は、通常のクロックに加えて、4 倍クロックを使用して、プログラム・メモリ 7 とデータ・メモリ 8 とに対するアクセスを行う。すなわち、図 8 に示されるように、プログラム・メモリ 7 及びデータ・メモリ 8 からのデータの読み込みには 4 倍クロックを使用し、A-REG 28 やデータ・メモリ 8 へのデータの書き込みにはシングルクロックを使用する。また、データ・メモリ 8 上のアドレスの計算処理をコマンド解釈処理から 2 分の 1 クロック遅らせる方式を採用している。これにより、NOP を挿入しなくても、ロード命令による A-REG 28 の更新が、ストア命令により A-REG 28 のデータをデータ・メモリ 8 に書き込むタイミング (図中の矢印 t 7 で示されるタイミング) に間に合うようになり、従って、CPU コア 2 の処理の高速化を図ることができる。

【0025】

次に、図9を参照して、このCPUコア2に採用されているマイクロ・コントローラ5を簡易な構成とするための工夫について説明する。このCPUコア2で用いられる命令51は、マイクロコード52、第1オペコード53、第2オペコード54、及びアドレス/即値データ55より構成される。この命令51の全体のレングスは32ビットであり、その先頭8ビットがマイクロコード52に割り当てられている。このマイクロコード52の各ビットのうちの先頭5ビット61～65は、データパス6中のレジスタやデータ・メモリ8の制御に用いられる制御ビットCである。マイクロコード52中の各ビット61～64は、それぞれA-REG28、データ・メモリ8、B-REG29、X-REG25に対応した書き込み制御用のビットであり、また、制御ビット65は、X-REG25に対するインクリメント指示用のビットである。図1で前述したマイクロコード・コントローラ9は、A-REG28、B-REG29、X-REG25のLD端子、及びデータ・メモリ8のWR端子との間にイネーブル信号出力用の制御線L1～L4を有しており、また、X-REG25のINC端子との間にインクリメント指示信号出力用の制御線L5を有している。マイクロコード・コントローラ9は、命令51中のマイクロコード52の読み込み時に、マイクロコード52中の各制御ビット61～64が1（オン）であれば、各制御線L1～L4を介して各制御ビット61～64のそれぞれに対応したレジスタ25、28、29及びデータ・メモリ8に書き込み許可用のイネーブル信号を送出するが、各制御ビット61～64が0（オフ）の場合には、イネーブル信号を送出しない。また、マイクロコード・コントローラ9は、マイクロコード52中の制御ビット65が1（オン）であれば、制御線L5を介してX-REG25にインクリメント指示信号を出力するが、制御ビット65が0（オフ）の場合には、イネーブル信号を送出しない。これにより、マイクロコード52中の各制御ビット61～65のオン/オフ情報をそのまま用いて、レジスタ25、28、29及びデータ・メモリ8を制御することができるので、従来、マイクロ・コントローラが、命令の内容を解読した上で行っていた各種レジスタやメモリに対する制御の一部を、命令の内容を解読することなく、容易に行うことができる。従って、マイクロ・コントローラ5が行う制御処理から主要なレジスタやメモリに対する制御処理を除去すること

ができるので、マイクロ・コントローラ 5 の構成を簡易にして、図 1 の制御部 20 全体の構成の簡易化を図ることができる。

【0026】

上記の CPU コア 2 は、従来の ASIC 等に組み込まれたシステム LSI 上の CPU コアとは異なり、システム LSI を構成する IP の仕様に応じて、CPU コア 2 のアーキテクチャをカスタマイズすることができ、また、変更した内容に応じて CPU コア 2 のテストを行うことができる。この CPU コア 2 のカスタマイズ及びテストは、パソコンを用いて行う。すなわち、CPU コア 2 のカスタマイズに先立って、図 11 に示される CD-ROM 73（請求項でいうコンピュータ読み取り可能な記録媒体）からパソコン 71 に、CPU コア 2 のソースと、CPU コア 2 のカスタマイズ及びテストを行うためのツールをインストールしておき、このツールを用いて CPU コア 2 のカスタマイズ及びテストを行う。

【0027】

図 10 に上記の CPU コア 2 のカスタマイズ処理とカスタマイズ後の CPU コア 2 の検証処理の流れを示す。ユーザは、ターゲットとなる IP 3 に応じて、CPU コア 2 の命令の追加、変更、削除の仕様を決定し（#1）、この仕様に応じて、図 12 に示されるカスタマイズ画面 81 から CPU コア 2 のアーキテクチャの定義を行う（#2）。具体的には、カスタマイズ画面 81 上のレジスタ定義用ウィンド 82 から CPU コア 2 内の各種レジスタの追加、変更、削除を行い、また、命令定義用ウィンド 83 から簡易言語を用いて CPU コア 2 内の各種命令の追加、変更、削除を行う。そして、プログラム・メモリ 7 に格納するプログラムを作成して（#3）、パソコン 71 上でこのプログラムのシミュレーション（オフライン・デバッグ）を行い（#4）、このシミュレーションによる検証結果が OK でなければ（#5 で NO）、検証結果が OK になるまで（#5 で YES）、プログラムの修正（#6）とシミュレーション（#4）を繰り返す。そして、検証結果が OK になると（#5 で YES）、#2 で定義した CPU コア 2 の内容に応じた CPU コア 2 の VHDL (VHSIC Hardware Description Language) レベルのソースの生成（#7）と、ターゲットとなる IP 3 の VHDL レベルのソースの生成（#8）を行った後、これらのソースの論理合成を行い（#9）、その論

理合成結果をFPGA1にダウン・ロードする（#10）。そして、論理合成結果をダウン・ロードした後のFPGA1とパソコン71とを接続して、FPGA1上のCPUコア2についての実機テスト（オンライン・デバッグ）を行う（#11）。

【0028】

次に、図11を参照して、上記の論理合成結果のダウン・ロード及びダウン・ロード後のCPUコア2の実機テストについて説明する。パソコン71は、インタフェース装置（ASAP: Adaptive Scan Agent Pod）72を介してFPGA1と接続される。このASAP72は、TAP（Test Access Port）と呼ばれるバウンダリ・スキャン・テスト用の専用端子を備えており、パソコン71は、この専用端子を介してFPGA1への論理合成結果のダウン・ロードや、FPGA1上のCPUコア2の実機テストを行う。FPGA1は、IP3やCPUコア2等のユーザ・ロジック回路のデバッグ用モジュールであるASH36を有している。ユーザは、パソコン71からASAP72及びASH36を介してFPGA1上のCPUコア2やIP3のレジスタやメモリに対するデータの読み書きを行うことにより、CPUコア2やIP3の実機テストを行うことができる。パソコン71とASAP72との間のデータ伝送は、RS-232Cインタフェースにより行われるが、ASAP72とASH36の間のデータ伝送は、バウンダリ・スキャン・テスト用のJTAGインタフェースにより行われる。

【0029】

上述したように、本実施形態によるFPGA1によれば、図9で前述したように、CPUコア2の命令がマイクロコード52を含み、マイクロコード・コントローラ9が、命令を読み込んだ後に、マイクロコードを構成する各ビットのオン／オフ情報を、制御線を介して各ビットに割り当てられたレジスタ25、28、29やデータ・メモリ8に伝達して、これらを制御するようにしたので、マイクロ・コントローラ5の行う処理を簡略化することができる。また、CPUコア2内のレジスタや命令を図12で前述したカスタマイズ画面81から削除・変更・追加することができるので、不要なレジスタ、命令等を削除したり、便利な命令を追加することにより、マイクロ・コントローラ5の構成やその処理の簡略化を

行うことができる。これにより、FPGA 1 上に簡易なCPUコア2を構築することができるので、従来のASICに用いられているCPUコアをFPGA 1に直接組み込んだ場合と比較して、FPGA 1 上におけるCPUコア2の占有面積を小さくすることができる。

【0030】

本発明は、上記実施形態に限られるものではなく、様々な変形が可能である。例えば、上記実施形態では、マイクロ・コントローラ5とは別にマイクロコード・コントローラ9を設けたが、マイクロコード・コントローラ9の機能をマイクロ・コントローラ5に組み込んで、これらを一体的に構成してもよい。また、上記実施形態では、通常のクロックに加えて4倍クロックを使用して、CPUコア2の処理の高速化を図ったが、使用するクロックの速さの組み合わせはこれに限らない。

【0031】

【発明の効果】

以上のように請求項1の発明によれば、制御部がCPUコアの保有するマイクロコードを含む命令を読み込み、この命令に含まれるマイクロコードを構成する各ビットのオン／オフ情報を制御線を介して各ビットに割り当てられたレジスタやメモリに伝達して、これらレジスタやメモリの状態を制御するようにしたので、制御部の行う処理を簡略化することができ、従って、制御部を簡易な構成とすることができると共に、フィールド・プログラマブル・ゲート・アレイ等の論理集積回路上に簡易なCPUコアを構築することができる。また、従来のASICに用いられているCPUコアをフィールド・プログラマブル・ゲート・アレイ等の論理集積回路に直接組み込んだ場合と比較して、論理集積回路上におけるCPUコアの占有面積を小さくすることができる。さらにまた、従来の解析ロジックに相当するハードウェア回路を作成する方式のフィールド・プログラマブル・ゲート・アレイ等の論理集積回路に比べて、解析ロジックの省スペース化を図ることができ、ロジックを更新した場合でも、ハードウェア回路を組み替えることなく、CPUコア内のプログラムの入れ替えだけで対応することができる。

【0032】

また、制御部とアドレスポインタ用レジスタとの間にインクリメント又はデクリメント指示用の制御線を設けて、アドレスポインタ用レジスタが、この制御線を介して制御部からのインクリメント又はデクリメント指示用のビットのオン情報を受け取った場合に、このレジスタの保持しているアドレスの値をカウント・アップ又はカウント・ダウンさせることにより、このアドレスポインタ用レジスタを用いて、メモリ上の連続したアドレス空間のデータ処理を効率的に行うことができる。

【 0 0 3 3 】

また、請求項 3 の発明によれば、制御部が通常のクロックよりも高速のクロックを使用して、完全同期式メモリであるプログラム格納用メモリ及びデータ格納用メモリからのデータを読み込むようにしたので、通常のクロックを使用してこれらのデータを読み込んだ場合と比べて、これらのメモリに格納された命令やデータの読み込みのタイミングを早めることができる。従って、例えば、データ格納用メモリからレジスタへのデータの読み出し命令と、同じレジスタからデータ格納用メモリへのデータの書き込み命令が連続して実行された場合でも、データ格納用メモリからレジスタへのデータの読み出しを、レジスタからデータ格納用メモリへのデータの書き込みに間に合わせることもできる。これにより、3 段パイプライン構成上で完全同期式メモリを採用しているにも拘わらず、パイプライン・ストールが生じないようにすることができ、従って、CPU コアの処理の高速化を図ることができる。

【 0 0 3 4 】

また、CPU コアが、メモリデータ入出力用のマルチプレクサを介することなく、汎用レジスタへのデータ入出用のマルチプレクサのみを介して汎用レジスタとメモリとを接続するデータ伝送経路を持ち、このデータ伝送経路を使用してメモリからのデータを汎用レジスタに入力することにより、従来のメモリデータ入出力用のマルチプレクサと汎用レジスタへのデータ入出用のマルチプレクサの 2 段のマルチプレクサを介してメモリから汎用レジスタへのデータの読み込みを行う CPU コアと比べて、汎用レジスタへのデータの読み込み処理の高速化を図ることができる。

【 0 0 3 5 】

また、CPUコアが、ALUを介することなく、メモリデータ入出力用のマルチプレクサのみを介して汎用レジスタとメモリとを接続するデータ伝送経路を持ち、このデータ伝送経路を使用してメモリへのデータを出力することにより、従来のALUとメモリデータ入出力用のマルチプレクサを介して汎用レジスタからメモリへのデータの書き込みを行うCPUコアと比べて、汎用レジスタからメモリへのデータの書き込み処理の高速化を図ることができる。

【 0 0 3 6 】

また、CPUコアは、データの入力部と出力部が独立してアクセス可能なメモリであるFIFOからのデータ入力用のインタフェースが組み込まれたマルチプレクサを有し、このマルチプレクサを介してFIFOからのデータを読み込むための命令を持つことにより、FIFOのデータを容易にCPUコア内に読み込むことができる。これにより、CPUコアが、FIFOを介して論理集積回路内の異なるIP間のデータ転送を容易に行うことができる。

【 0 0 3 7 】

また、請求項7の発明によれば、コンピュータにソースを読み取らせることにより、上記に記載の発明と同等の効果を得ることができる。

【図面の簡単な説明】

【図1】 本発明の一実施形態による論理集積回路上のCPUコア周辺の構成を示す図である。

【図2】 上記CPUコア内のデータ・メモリから汎用レジスタにデータをロードする際の配線経路に関する説明図である。

【図3】 上記CPUコア内の汎用レジスタからデータ・メモリにデータをストアする際の配線経路に関する説明図である。

【図4】 上記CPUコアにおいてFIFOからのデータ入力命令を実現するための機構の説明図である。

【図5】 上記CPUコアにおいてポストインクリメント命令を実現するための機構の説明図である。

【図6】 従来の3段パイプライン構成上で完全同期式メモリを採用したC

P Uコアでロード命令とストア命令を続けて実行した場合におけるパイプライン制御の問題点の説明図である。

【図 7】 上記図 6 の説明図中におけるパイプライン制御の問題点をロード命令とストア命令の間にNOPを挿入することにより解決した場合におけるパイプライン制御の説明図である。

【図 8】 上記図 6 の説明図中におけるパイプライン制御の問題点を通常のクロックに加えて4倍クロックを用いてメモリにアクセスすることにより解決した場合におけるパイプライン制御の説明図である。

【図 9】 上記CPUコアの制御部をマイクロコードを用いて簡易な構成とするための説明図である。

【図 10】 上記CPUコアのカスタマイズとデバッグの処理を示すフローチャートである。

【図 11】 上記FPGA上への論理合成結果のダウン・ロードとCPUコアの実機テストを行うための構成図である。

【図 12】 上記CPUコアのカスタマイズ用の画面を示す図である。

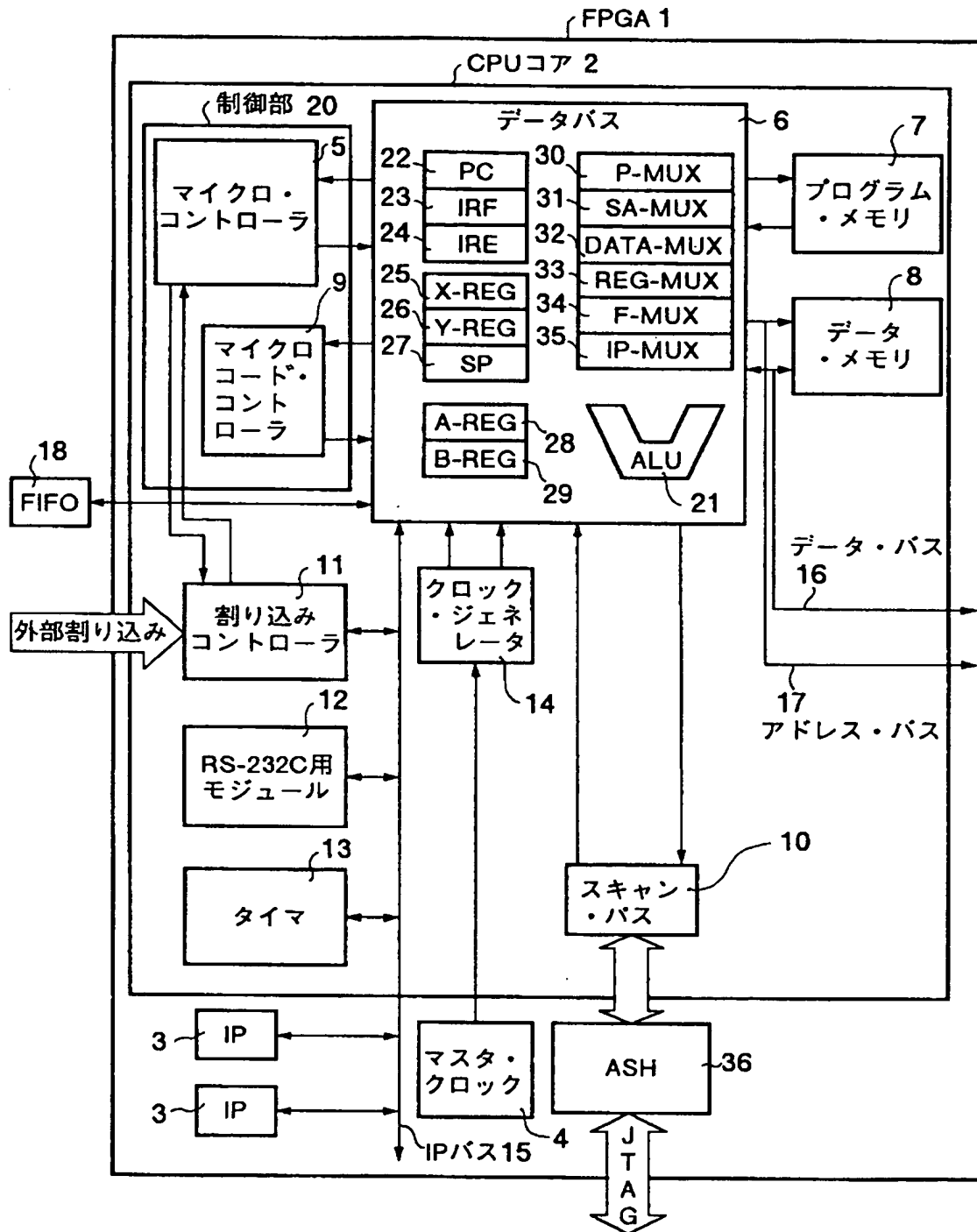
【符号の説明】

- 1 F P G A (フィールド・プログラマブル・ゲート・アレイ)
- 2 C P U コア
- 7 プログラム・メモリ (プログラム格納用メモリ)
- 8 データ・メモリ (メモリ、データ格納用メモリ)
- 9 マイクロコード・コントローラ (レジスタ及びメモリの制御部)
- 1 8 F I F O
- 2 0 制御部 (全体の制御部)
- 2 1 A L U
- 2 5 X-REG (レジスタ、アドレスポインタ用レジスタ)
- 2 6 Y-REG (アドレスポインタ用レジスタ)
- 2 8 A-REG (レジスタ、汎用レジスタ)
- 2 9 B-REG (レジスタ)
- 3 2 DATA-MUX (メモリデータ入出力用のマルチプレクサ)

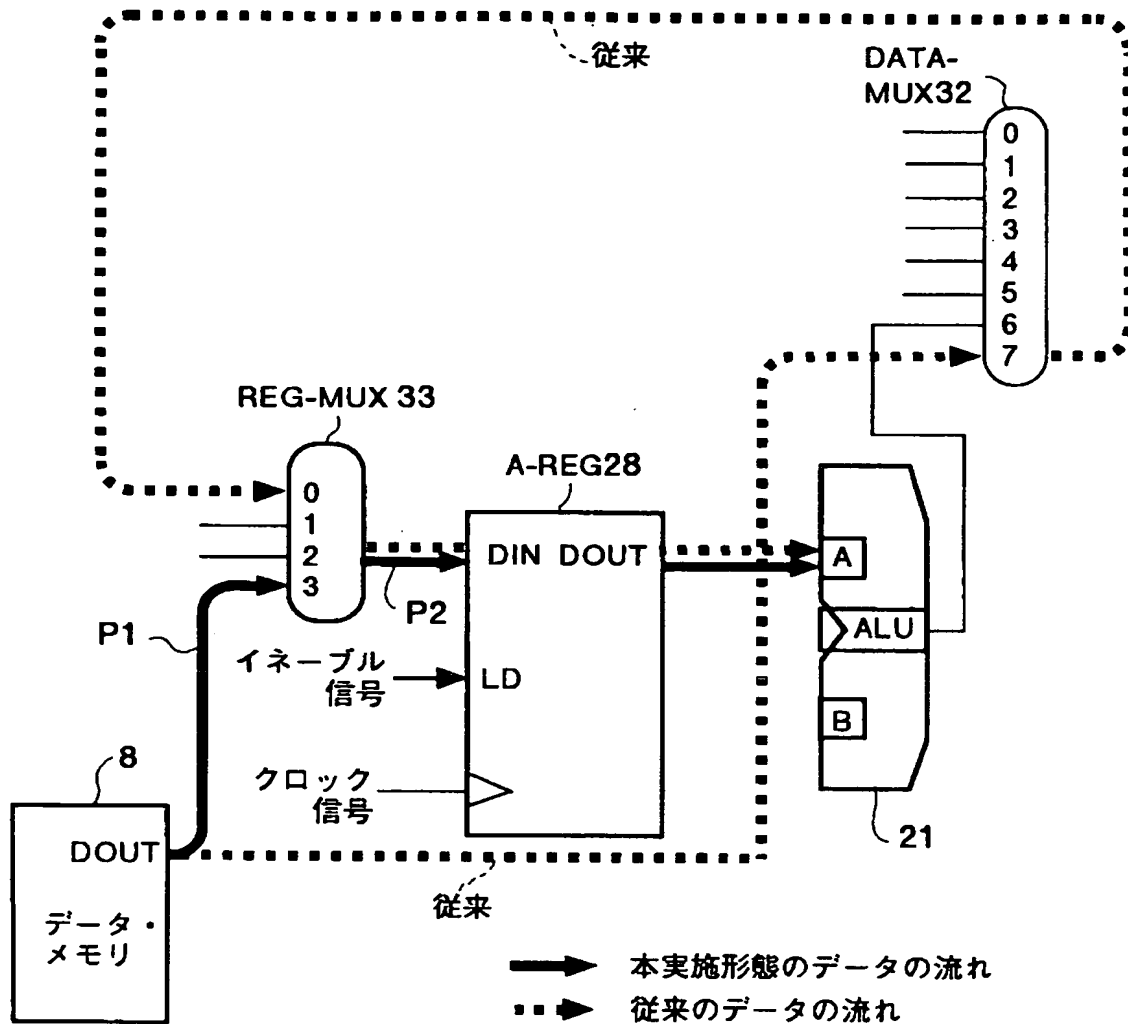
- 3 3 R E G - M U X (汎用レジスタ用のマルチプレクサ)
- 3 4 F - M U X (F I F O用のインタフェースが組み込まれたマルチプレクサ)
- 7 3 C D - R O M (コンピュータ読み取り可能な記録媒体)
- L 1 ~ L 5 制御線 (イネーブル信号出力用の制御線)
- L 5 制御線 (指示用制御線)
- P 1 , P 2 配線 (請求項 4 におけるデータ伝送経路を構成する配線)
- P 3 , P 4 配線 (請求項 5 におけるデータ伝送経路を構成する配線)

【書類名】 図面

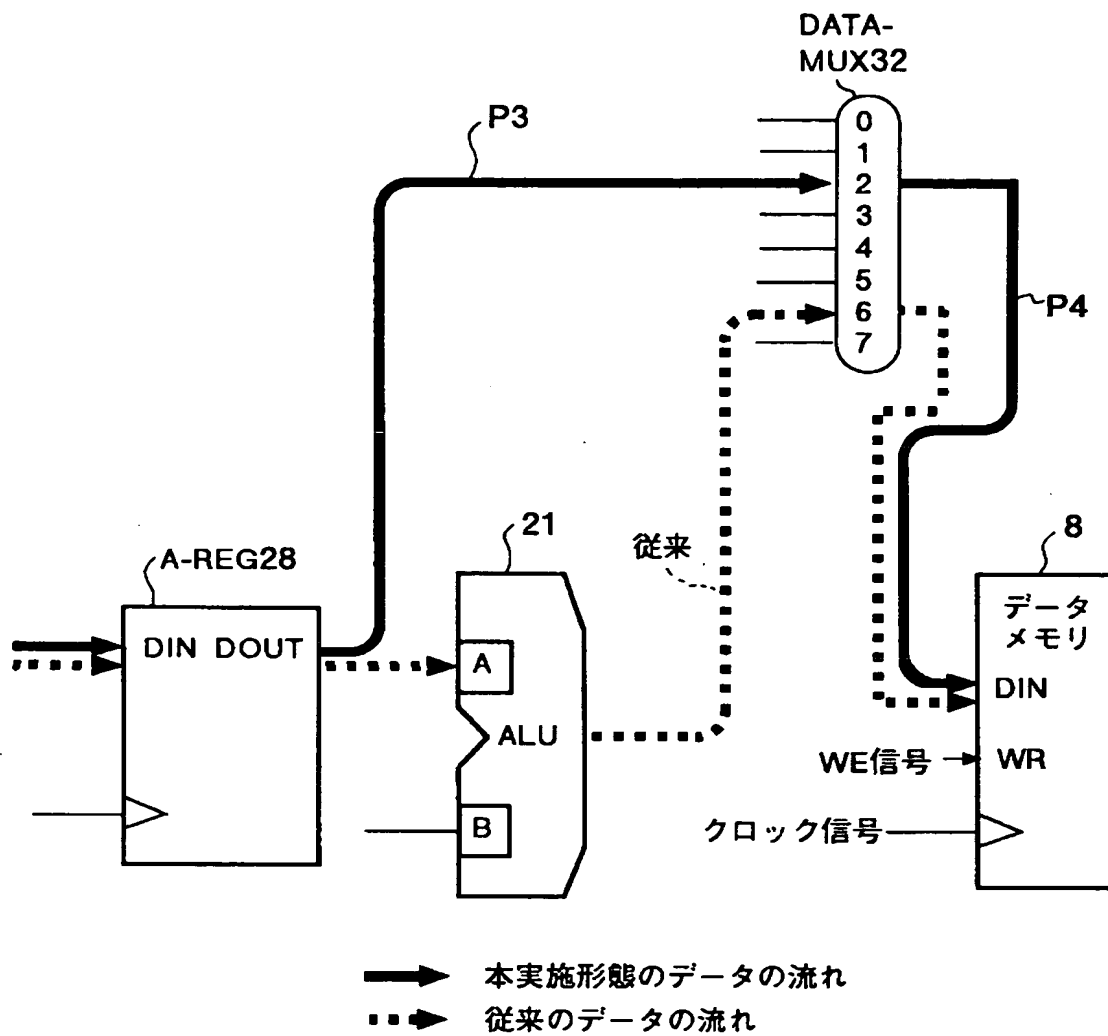
【図 1】



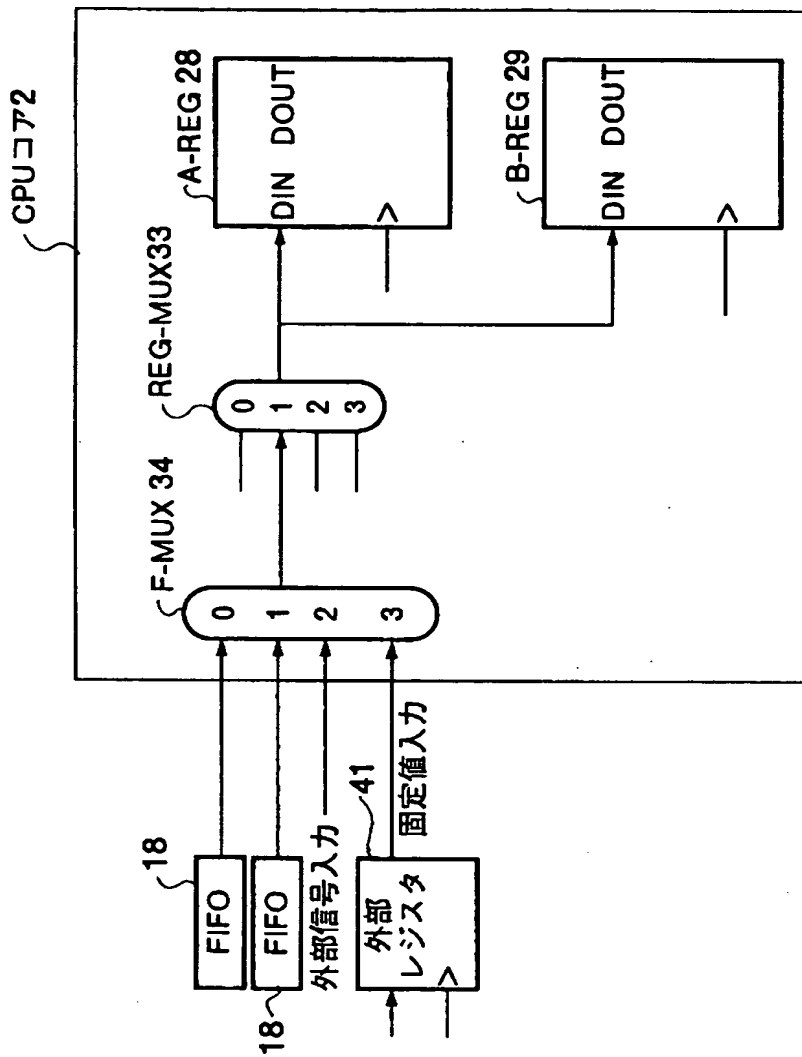
【図 2】



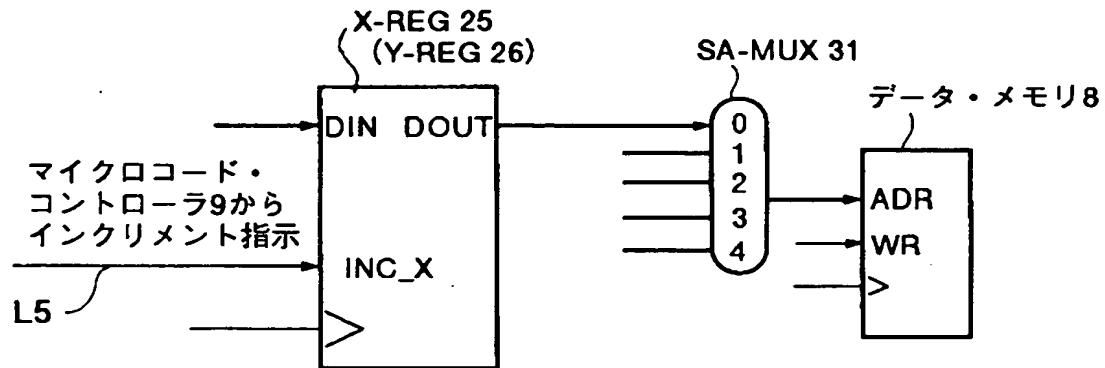
【図 3】



【図 4】

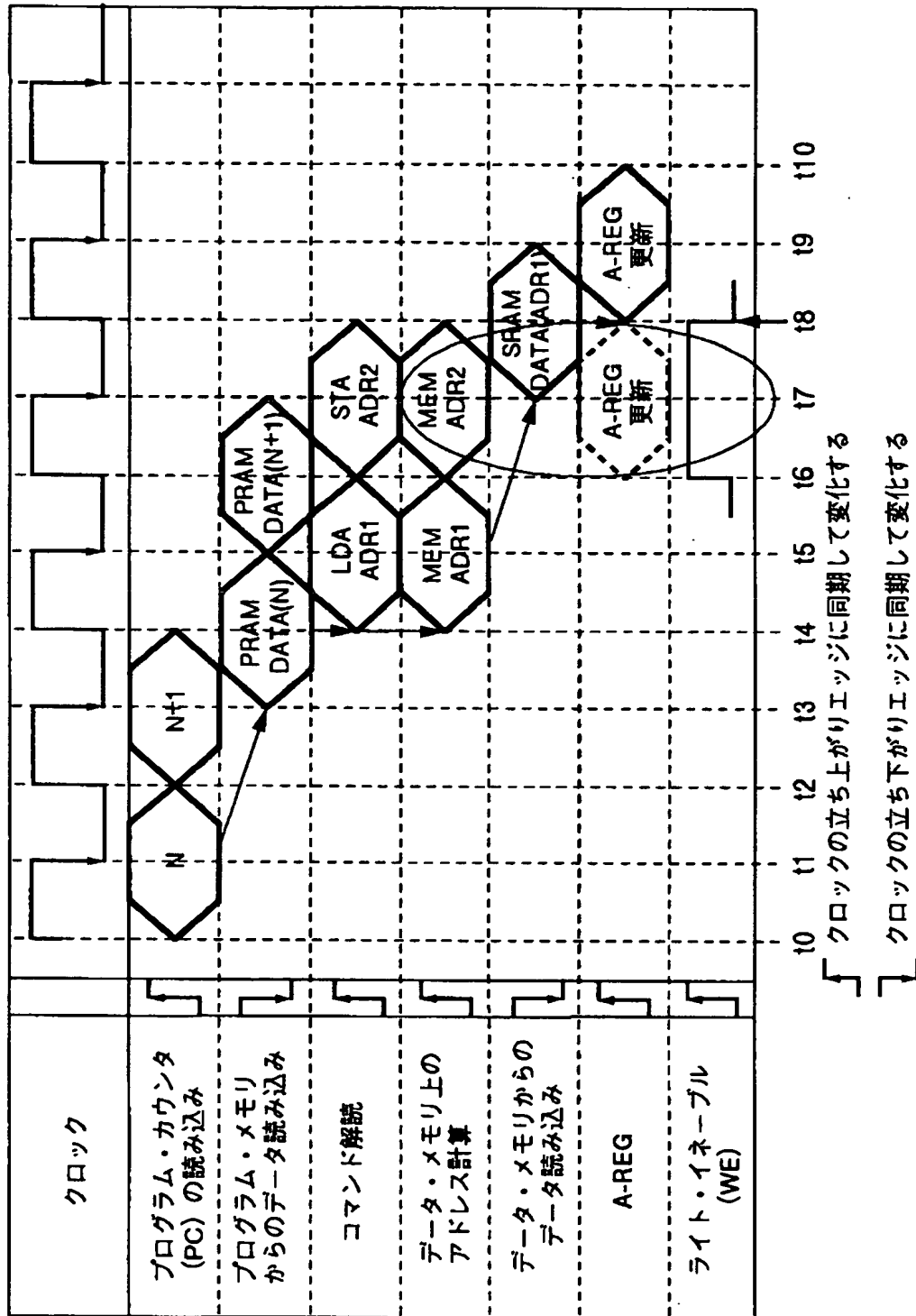


【図 5】



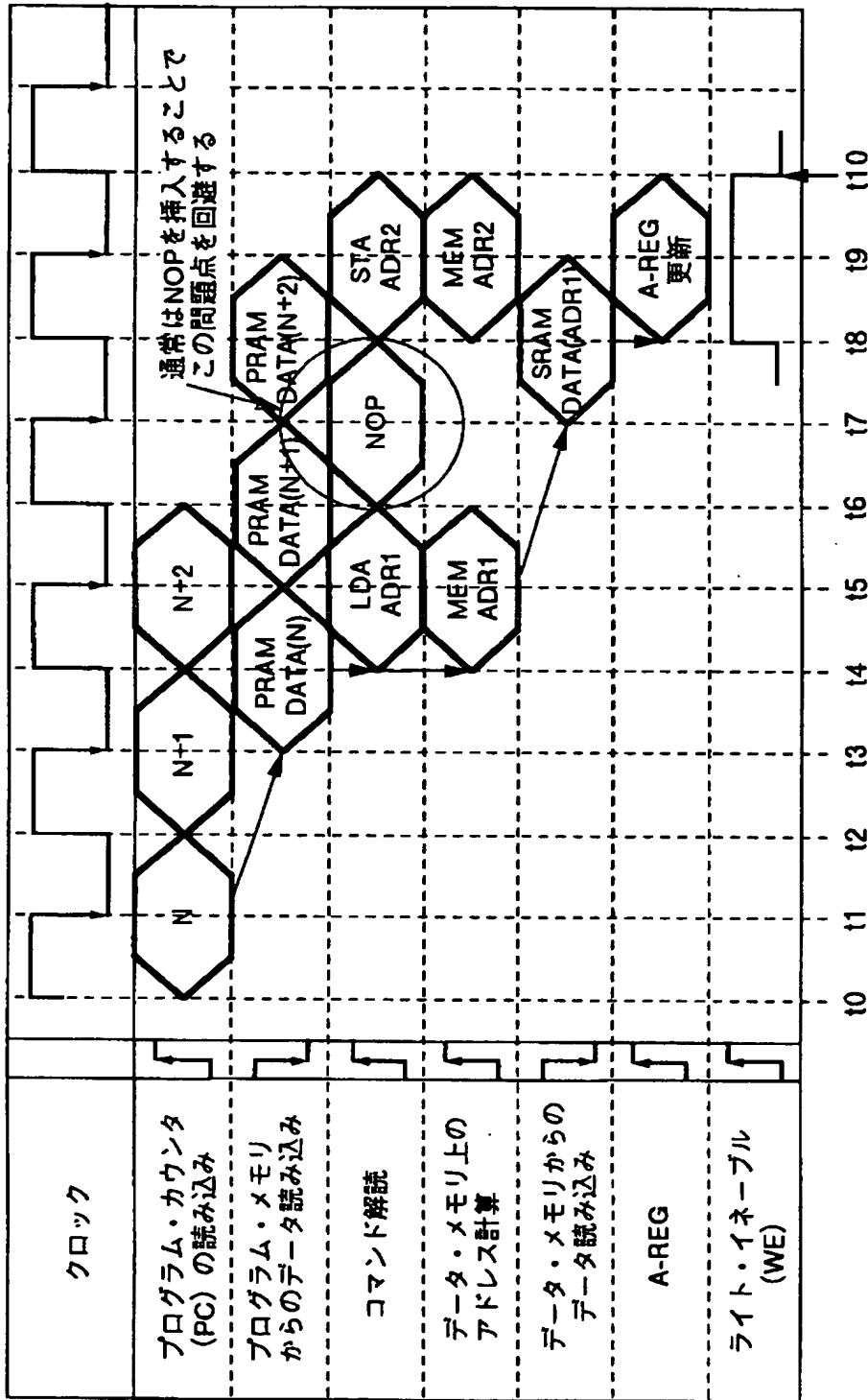
【図 6】

シングルクロックでのコントロール例 (LDA&STA連続コマンドの時)



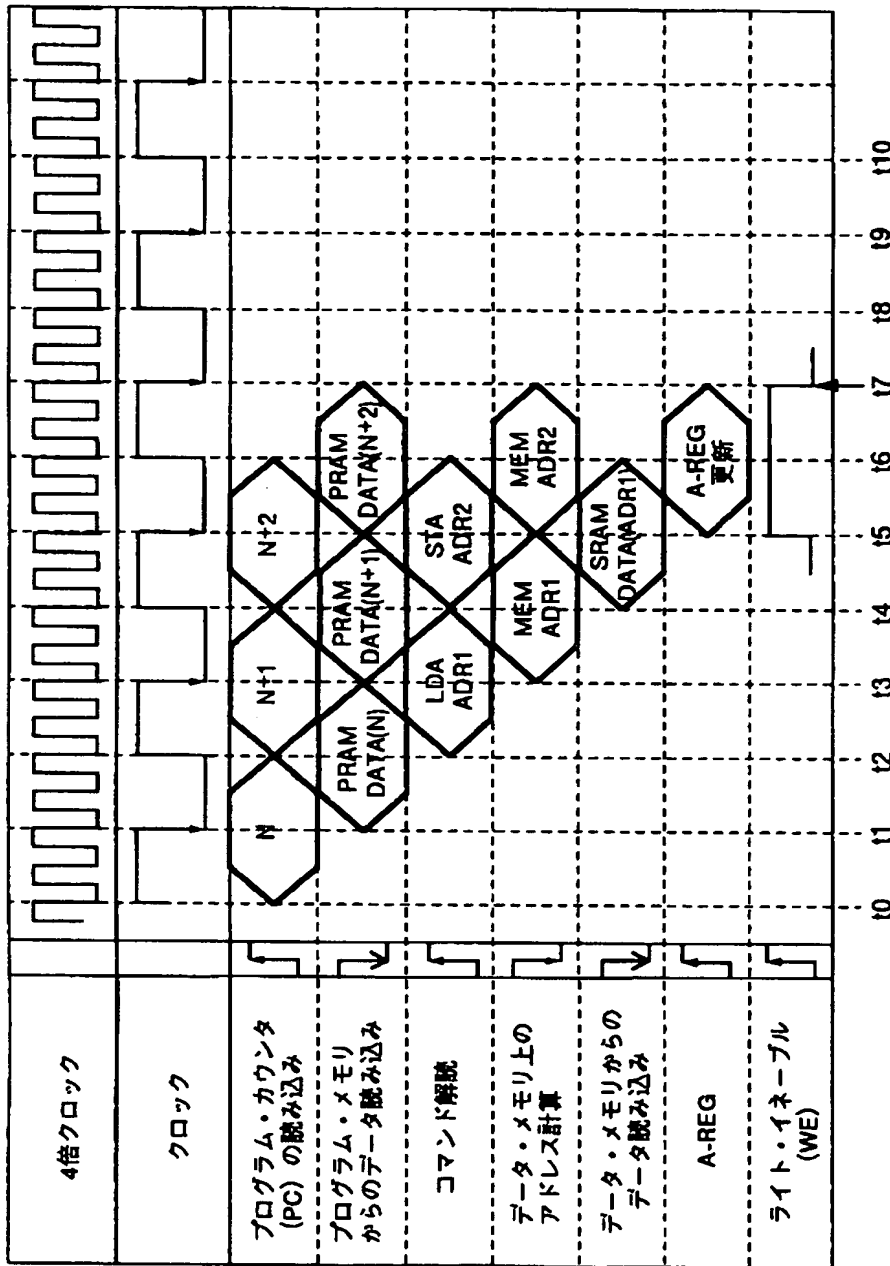
【図 7】

NOP を挿入して解決する例



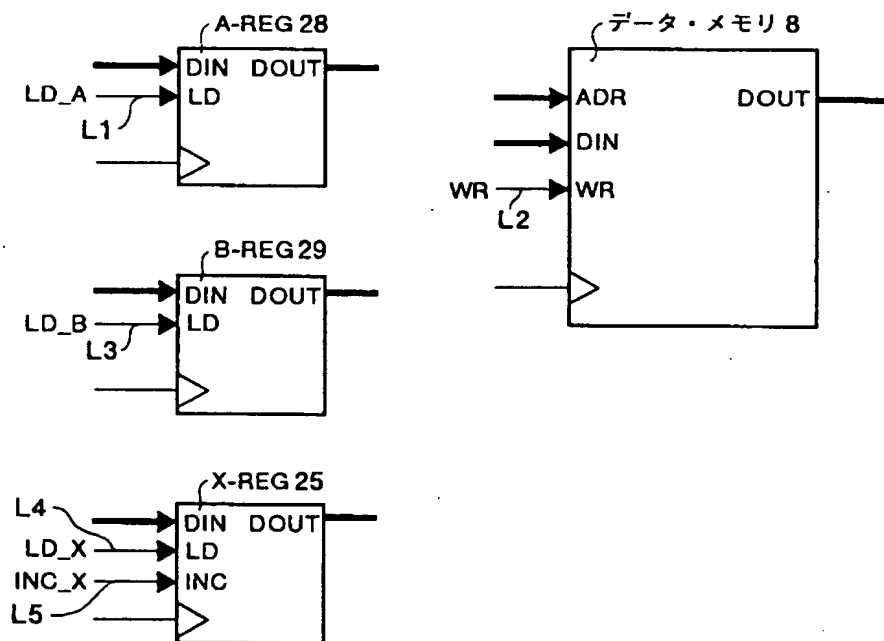
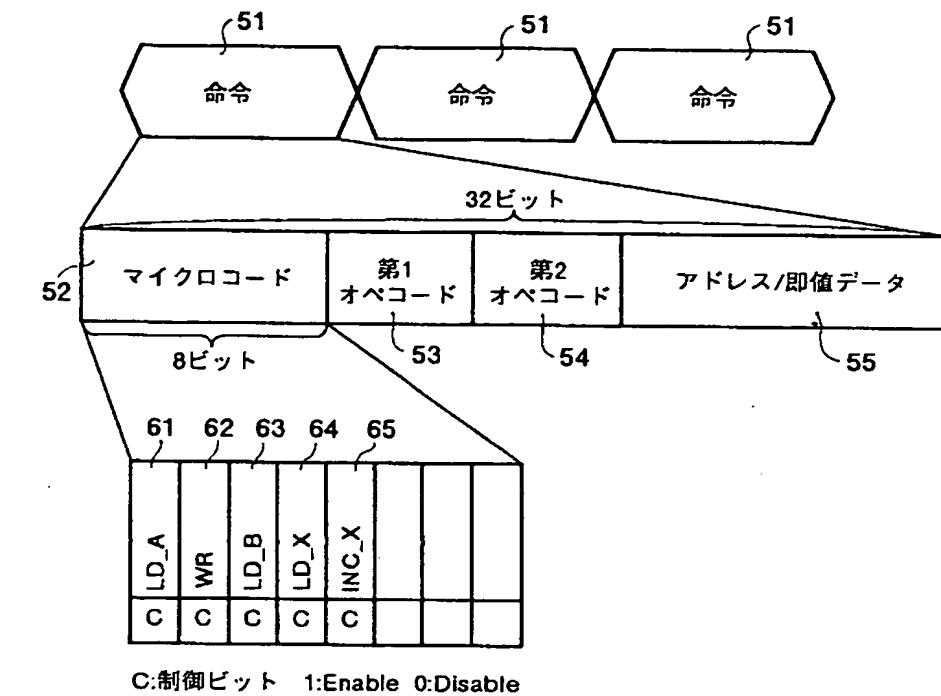
【図 8】

本実施形態のCPUコアのコントロール方式

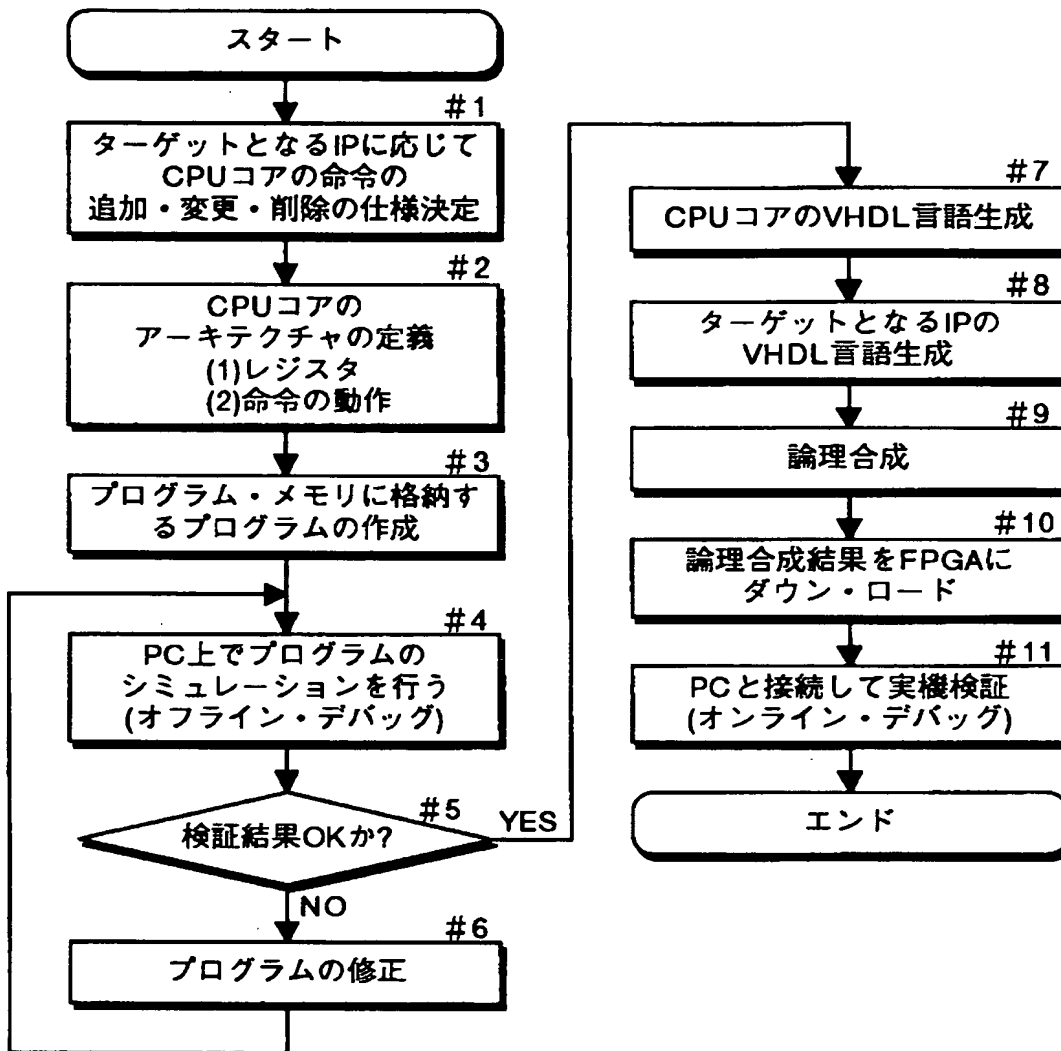


- ┌ クロックの立ち上がりエッジに同期して変化する
- └ クロックの立ち下がりエッジに同期して変化する
- └ 4倍クロックの立ち下がりエッジに同期して変化する

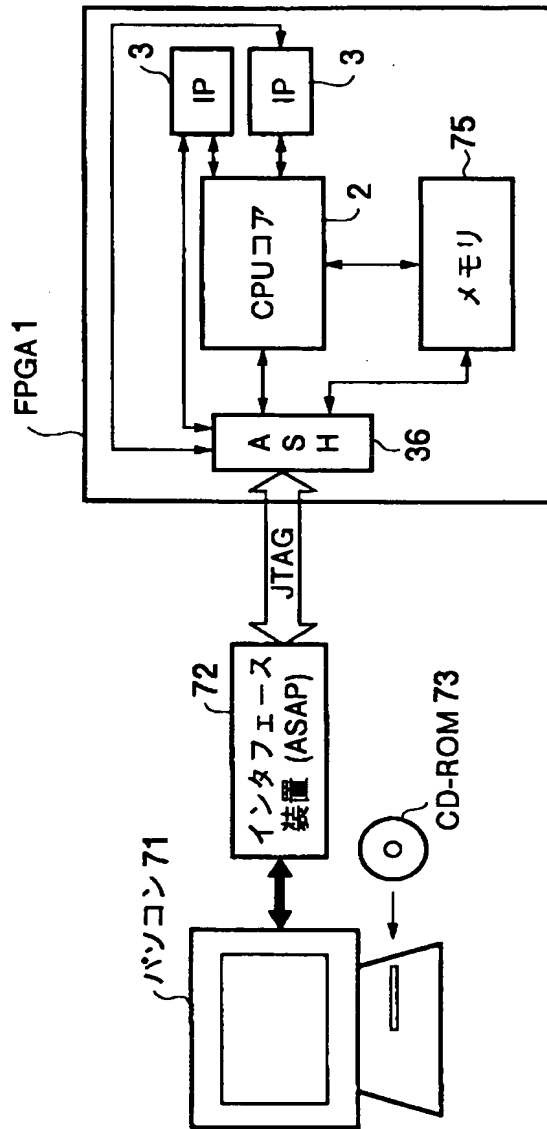
【図 9】



【図 1 0】



【図 1 1】



【図 12】

81
~

COUコアのカスタマイズ

レジスタ (R)

レジスタ名 ビット長 初期値 レジスタ種類

a	16	h"0000"	汎用レジスタ
b	16	h"0000"	汎用レジスタ
z	1	b"0"	ゼロ・フラグ
.	.	.	.
.	.	.	.

追加 設定
編集 キャンセル
削除 ヘルプ

命令 (I)

命令名 マイクロコード 動作定義

bge	h"9815FFFF"	"if(\$1==0 or \$z==1)\$pc: =op[11:0]: "
emm	h"10A50000"	"\$msk: =op[3:0]: "
.	.	.
.	.	.
.	.	.

追加 編集 削除

レジスタの定義

命令の動作定義

【書類名】 要約書

【要約】

【課題】 F P G A等の論理集積回路上において、簡易で高パフォーマンスの C P Uコアを構築することができるようにして、解析ロジックの省スペース化を図ると共に、ロジック更新の度にハードウェア回路を組み替える必要をなくす。

【解決手段】 制御部が、命令を読み込み、この命令に含まれるマイクロコードを構成する各ビットのオン／オフ情報を、制御線 L 1 ～ L 5 を介して各ビットに割り当てられたレジスタ 2 5, 2 8, 2 9 やデータ・メモリ 8 に直接伝達して、これらを制御するようにしたので、制御部の行う処理を簡略化することができる。これにより、制御部を簡易な構成とすることができるので、F P G A等の論理集積回路上に簡易な C P Uコアを構築することができる。

【選択図】 図 9

出 願 人 履 歴 情 報

識別番号 [595107944]

1. 変更年月日 1995年 7月 4日

[変更理由] 新規登録

住 所 大阪市北区東天満1丁目4番16号 都市計画合同ビル2F
氏 名 株式会社ローラン

出 願 人 履 歴 情 報

識別番号 [000000262]

1. 変更年月日	1990年 8月18日
[変更理由]	新規登録
住 所	大阪府大阪市淀川区田川2丁目1番11号
氏 名	株式会社ダイヘン